

Podstawy sztucznej inteligencji

Kinga Synowiec, Inżynieria obliczeniowa, grupa 2

Sprawozdanie nr 2

- **Temat ćwiczenia:**

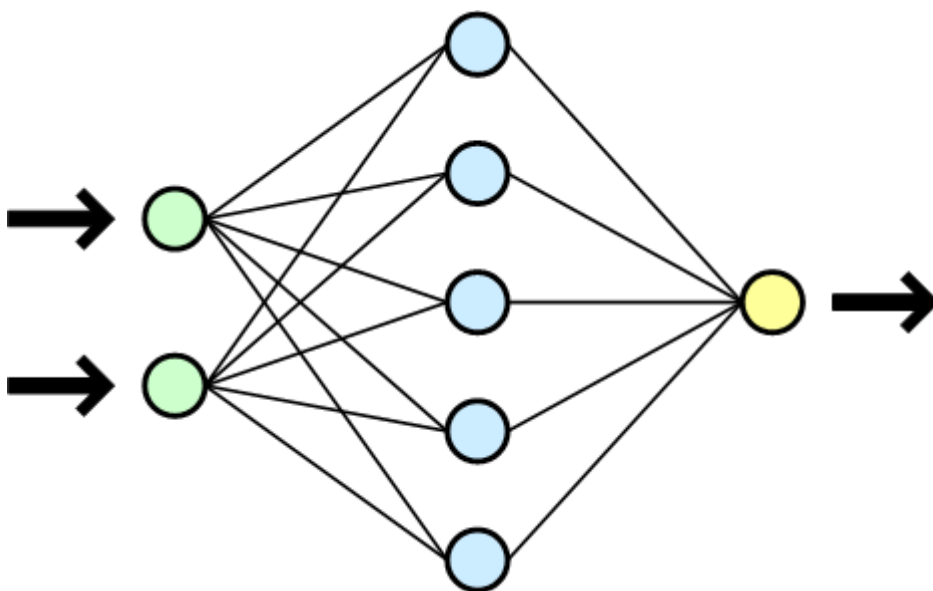
Budowa i działanie sieci jednowarstwowej.

- **Cel ćwiczenia:**

Poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter.

- **Wstęp teoretyczny:**

Sieć neuronowa (sztuczna sieć neuronowa) – ogólna nazwa struktur matematycznych i ich programowych lub sprzętowych modeli, realizujących obliczenia lub przetwarzanie sygnałów poprzez rzędy elementów, zwanych sztucznymi neuronami, wykonujących pewną podstawową operację na swoim wejściu. Oryginalną inspiracją takiej struktury była budowa naturalnych neuronów, łączących je synaps, oraz układów nerwowych, w szczególności mózgu. Czasem nazwą sztuczne sieci neuronowe określa się interdyscyplinarną dziedzinę wiedzy zajmującą się konstrukcją, trenowaniem i badaniem możliwości tego rodzaju sieci.

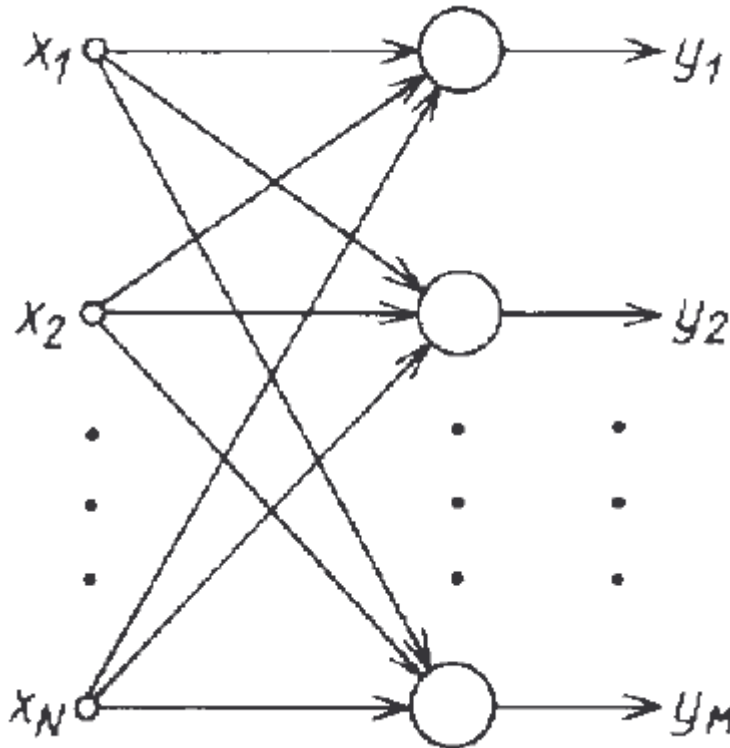


// źródło : https://pl.wikipedia.org/wiki/Sie%C4%87_neuronowa

Cechy sieci jednokierunkowej jednowarstwowej:

- neurony ułożone w jednej warstwie

- w węzłach wejściowych nie zachodzi żaden proces obliczeniowy, nie tworzą więc warstwy neuronów
- przepływ sygnałów jednokierunkowy: od warstwy wejściowej do wyjściowej
- najczęściej pełne połączenia (każdy węzeł z każdym neuronem)
- nazwa sieci określona przez sposób doboru wag oraz wybór metody uczenia (np. perceptron jednowarstwowy, sieć Kohonena itp.)



// źródło :

https://enauczanie.pg.edu.pl/moodle/pluginfile.php/135273/mod_resource/content/1/w07.pdf

- **Opis użytych funkcji i metod zaczerpniętych z pakietu MATLAB:**

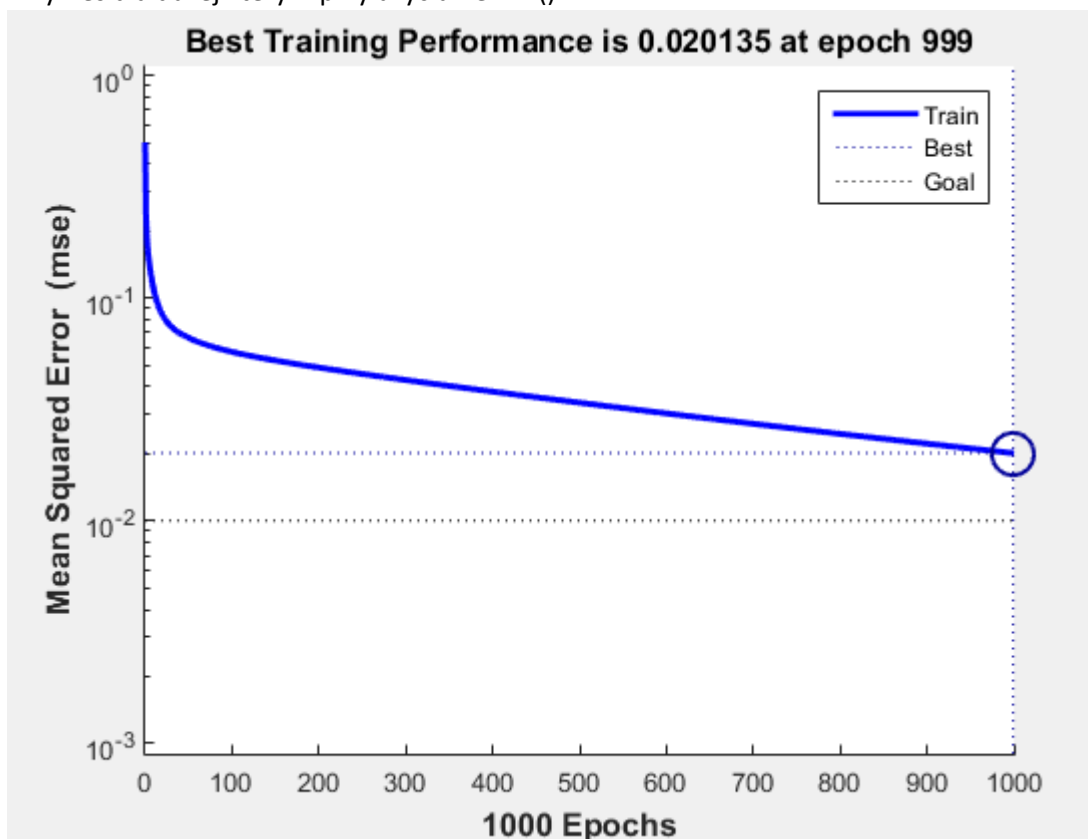
- net – zmienna, do której przypisywana jest nowa sieć neuronowa.
- newlin() – funkcja tworzy jednowarstwową sieć neuronową. Jako pierwszy argument przyjmuje wartości minimalne i maksymalne dla wejściowych elementów sieci dla funkcji tworzących sieć neuronową. Argument ten składa się z 25 par oznaczających odpowiednio 1, czyli wartość maksymalną, lub 0 dla wartości minimalnej. Drugim argumentem jest liczba elementów wektora wyjściowego sieci, równa 1. Jego zadaniem jest przechowanie informacji, czy litera jest mała czy duża.
- newp() – funkcja tworzy prostą sieć neuronową, pobiera dane wejściowe i zwraca perceptron. Przyjmuje te same argumenty co newlin().
- WEJŚCIE/WYJŚCIE – zmienne uczące oraz zmienne przechowujące dane wyjściowe odpowiadające danym uczącym.
- net.trainParam.x – funkcja umożliwiająca określenie parametrów treningu sieci.
- train() – uczenie sieci z wykorzystaniem danych wejściowych i wyjściowych.
- x_test – zmienne dla danych sterujących
- sim() – symulacja sieci, parametry to sieć oraz badana litera

- Wyniki zebrane w postaci tabel z uwzględnieniem zmieniającego się współczynnika uczenia dla wybranych dużych oraz małych liter, przy użyciu kolejno funkcji newlin() i newp():

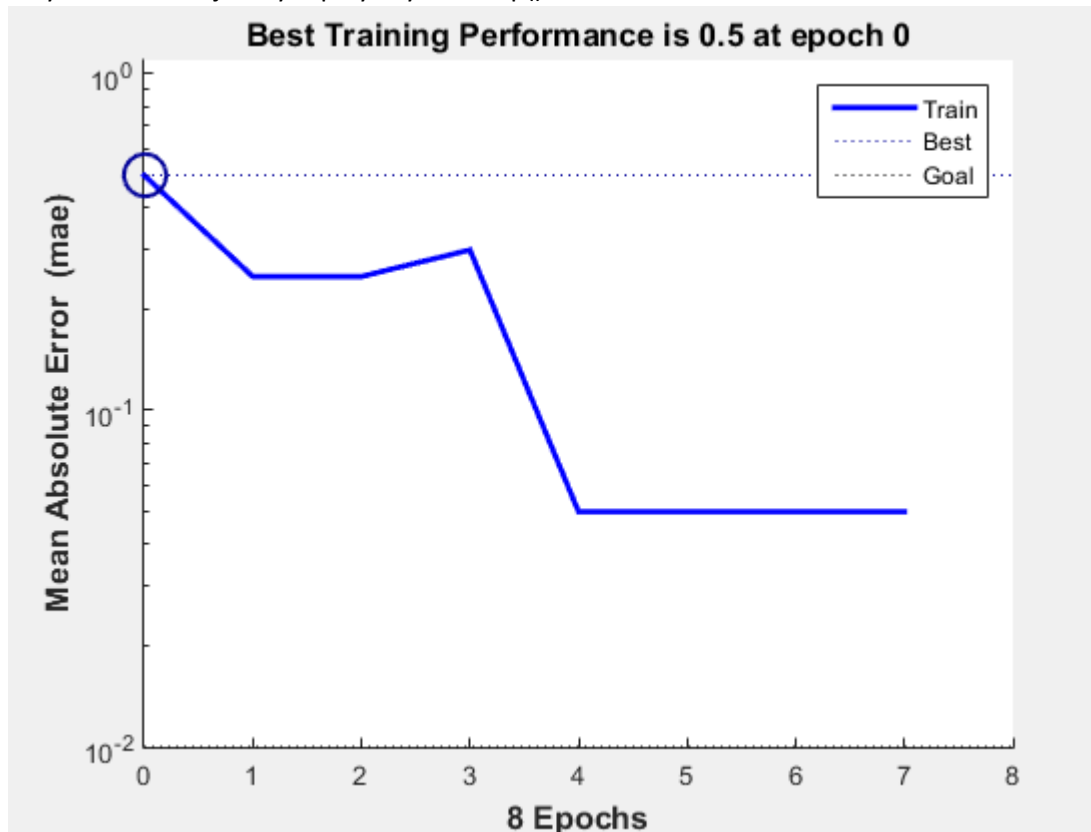
	0.001 - newlin	0.001 - newp	0.01 - newlin	0.01 - newp	0.1 - newlin	0.1 - newp
C	0.9629	1	0.9629	1	0.9629	1
k	0.0036	0	0.0036	0	0.0036	0
A	0.9356	1	0.9356	1	0.9356	1
a	0.0896	0	0.0896	0	0.0896	0
E	1.0372	1	1.0372	1	1.0372	1
e	0.0885	0	0.0885	0	0.0885	0
H	1.0269	1	1.0269	1	1.0269	1
h	0.0194	0	0.0194	0	0.0194	0

- Zrzuty ekranu przedstawiające przykładowe rezultaty po uruchomieniu programu:

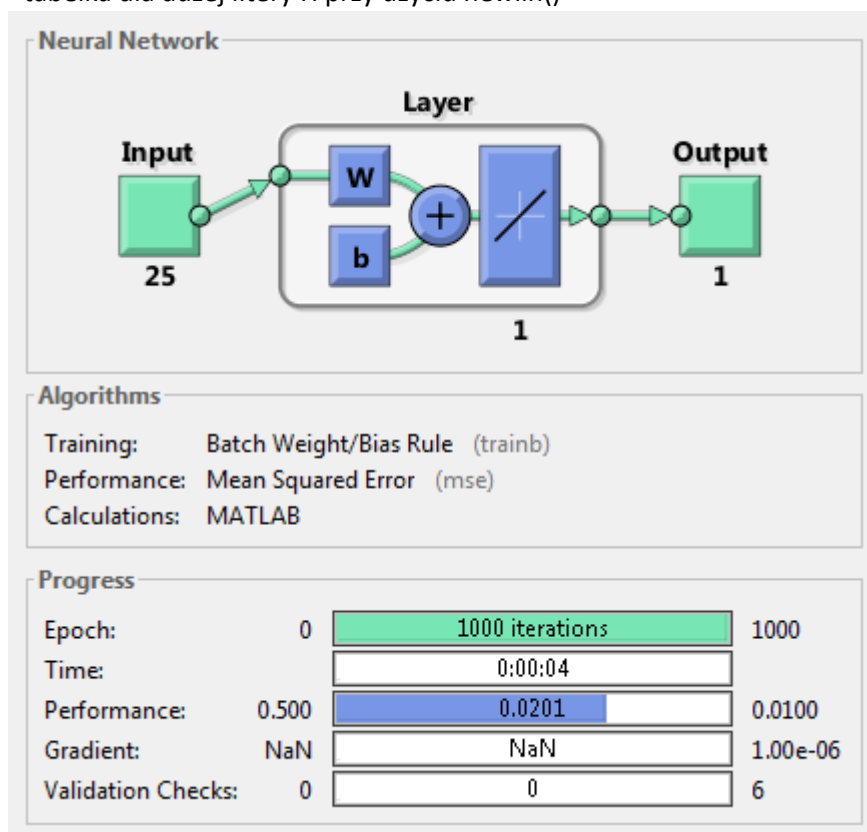
- wykres dla dużej litery H przy użyciu newlin()



- wykres dla dużej litery H przy użyciu newp()



- tabela dla dużej litery H przy użyciu newlin()



- Listing całego kodu programu wykonanego w MATLABie:

```
%%
close all; clear all; clc;

poczatek = [0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0
1;
            0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0
1;];
wyjscia = 1;
net = newlin(poczatek, wyjscia);
%net = newp(poczatek, wyjscia);

WEJSCIE = [0 0 1 1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1;
           1 1 1 0 1 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0;
           1 1 1 0 1 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0;
           1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0;
           0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
           1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1;
           0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
           0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0;
           0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0;
           1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
           1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1;
           1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0;
           1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 0 1 0;
           1 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0;
           1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
           1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
           0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0;
           0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0;
           0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0;
           1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
           1 0 1 1 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1;
           0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1;
           0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 1 1 1;
           0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1;
           1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0];

WYJSCIE = [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0];

net.trainParam.epochs = 1000;
net.trainParam.goal = 0.01;
net.trainParam.mu = 0.001;

net = train(net, WEJSCIE, WYJSCIE);

A_test = [0; 1; 1; 1; 0;
           1; 0; 0; 0; 1;
           1; 1; 1; 1; 1;
           1; 0; 0; 0; 1;
           1; 0; 0; 0; 1];
a_test = [0; 1; 1; 0; 0;
           0; 0; 0; 1; 0;
           0; 1; 1; 1; 0;
           1; 0; 0; 1; 0;
           0; 1; 1; 1; 1];
B_test = [1; 1; 1; 0; 0;
           1; 0; 0; 1; 0;
           1; 1; 1; 0; 0;
           1; 0; 0; 1; 0;];
```

```

      1; 1; 1; 0; 0];
b_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 1; 0;
          1; 1; 1; 0; 0];
C_test = [0; 1; 1; 1; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          0; 1; 1; 1; 0];
c_test = [0; 0; 0; 0; 0;
          0; 0; 0; 0; 0;
          0; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          0; 1; 1; 0; 0];
D_test = [1; 1; 1; 0; 0;
          1; 0; 0; 1; 0;
          1; 0; 0; 1; 0;
          1; 0; 0; 1; 0;
          1; 1; 1; 0; 0];
d_test = [0; 0; 0; 1; 0;
          0; 0; 0; 1; 0;
          0; 1; 1; 1; 0;
          1; 0; 0; 1; 0;
          0; 1; 1; 1; 0];
E_test = [1; 1; 1; 1; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 1; 0];
e_test = [0; 1; 1; 0; 0;
          1; 0; 0; 1; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          0; 1; 1; 0; 0];
F_test = [1; 1; 1; 1; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0];
f_test = [0; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0];
H_test = [1; 0; 0; 0; 1;
          1; 0; 0; 0; 1;
          1; 1; 1; 1; 1;
          1; 0; 0; 0; 1;
          1; 0; 0; 0; 1];
h_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 1; 1; 0; 0;
          1; 0; 1; 0; 0;
          1; 0; 1; 0; 0];
I_test = [1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0;
          1; 0; 0; 0; 0];

```

```

i_test = [1; 0; 0; 0; 0; 0;
          0; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0];
K_test = [1; 0; 0; 1; 0; 0;
          1; 0; 1; 0; 0; 0;
          1; 1; 0; 0; 0; 0;
          1; 0; 1; 0; 0; 0;
          1; 0; 0; 1; 0; 0];
k_test = [1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 0; 1; 0; 0; 0;
          1; 1; 0; 0; 0; 0;
          1; 0; 1; 0; 0; 0];
L_test = [1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 1; 1; 1; 0; 0];
l_test = [1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 0; 0; 0; 0; 0;
          1; 1; 1; 0; 0; 0];
testuj = sim(net, A_test)
if round(testuj) == 0
disp('Mala litera');
else
disp('Duza litera');
end

```

- **Wnioski:**

- Wartość współczynnika uczenia, mimo zmian jego wartości od 0.001 do 0.1 nie ma żadnego wpływu na otrzymane wyniki.
- Czas uczenia neutronu również nie zależy od współczynnika.
- Przy używaniu funkcji newp() liczba iteracji nie wynosi więcej niż 8, więc jest ona znacznie niższa niż w przypadku użycia funkcji newlin().
- Używanie funkcji newp() dawało bardzo konkretne wyniki – zawsze 1 w przypadku dużej litery i 0 w przypadku małej.
- Przy stosowaniu funkcji newlin() wyniki mocno się wahały, najmniejsze wyniki miały aż dwa 0 po przecinku przed wartością znaczącą, niektóre wyniki jednak przekraczały wartość 1.
- Sugerując się ilością iteracji można wnioskować, że program wykonuje się szybko.
- Na podstawie zaobserwowanych wyników można stwierdzić, że nie istnieje metoda pozwalająca na jednoznaczne i pewne zidentyfikowanie właściwego wyniku, dlatego zawsze należy dobrać metodę zgodnie z potrzebami i przeprowadzić różne próby, a najlepiej wykorzystać różne metody.